PATENT ABSTRACTS OF JAPAN

(11)Publication number : **05-274385**

(43)Date of publication of application : **22.10.1993**

(51)Int.Cl. G06F 15/60
G06F 9/46
G06F 11/22
G06F 11/26

(21)Application number : **04-066527**

(22)Date of filing : **25.03.1992**

(71)Applicant : **HITACHI LTD**

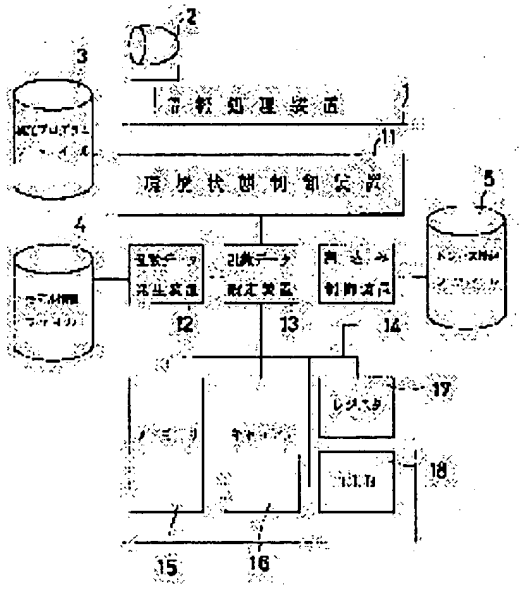
(72)Inventor : **SUZUKI KAORU**

(54) LOGIC VERIFICATION ENVIRONMENT CONTROLLING DEVICE

(57)Abstract:

PURPOSE: To provide the logic verification environment controlling device realizing accurate logic verification by setting environment equivalent to the real use environment and readily analyzing a fault in case of occurring a fault.

CONSTITUTION: The system consists of an information processor 1 for logic verification, system console 2, verification program file 3, model information file 4, and trace information file 5. The information processor 1 is provided with an environment state controlling device 11, random data generating device 12 for an entry embeding random number data, entry random number data setting device 13, interrupt controlling device 14 performing a back trace at the run away of a program, memory 15, cache 16, register 17, and TLB 18. Before executing the logic verification program, the resource in the information processor 1 is initialized.



LEGAL STATUS

[Date of request for examination] 05.09.1995

[Date of sending the examiner's decision of rejection] 28.04.1998

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] The random-number data generator which is a logic verification environment control unit to the logic simulation equipment which carries out circuit simulation of gate level, or an information processor, and creates random random-number data, The random-number data setting device which sets these random-number data as the data storage means in said logic simulation equipment or an information processor, The interrupt control equipment which controls the interruption generated within said logic simulation equipment or an information processor, Control of operator interface control, said logic simulation equipment, or an information processor, And the logic verification environment control unit characterized by having the environment condition control unit which carries out the edit output of trace information, and setting said logic simulation equipment or the resource in an information processor as an environment equivalent to a real operating environment before logic verification program execution.
[Claim 2] The logic verification environment control unit according to claim 1 characterized by creating so that these entry data may be adjusted in the logical format of said logic simulation equipment or an information processor, and embedding said random-number data to these entry data in case the entry data of said data storage means are created.
[Claim 3] The logic verification environment control unit according to claim 1 characterized by said random-number data generator generating the instruction and address data which operate on said specific logic simulation equipment or an information processor.
[Claim 4] The logic verification environment control unit according to claim 1 characterized by generating an interruption generating instruction operand code in dozens to hundreds of byte pitch in case said random-number data generator creates the random-number data set as said data storage means.
[Claim 5] The logic verification environment control unit according to claim 1 characterized by for said interrupt-processing equipment checking the interruption generating address and an interrupt type, carrying out back trace in the range traceable from the interruption generating address, and saving trace information.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Industrial Application] About the logic verification technique of the logical circuit which constitutes logic simulation equipment or an information processor, especially, this invention is complicated, is applied to the suitable logic verification environment control unit for highly precise verification of the big logical circuit of a scale, and relates to an effective technique.
[0002]
[Description of the Prior Art] For example, by the conventional logic verification method, from the condition similar to the condition that the memory in an information processor after power on reset, a cache, a register, TLB (Table Look-aside Buffer), etc. were initialized (zero clear), or it, the information processor set the logic verification program to a memory load or the memory which corresponds verification program data, the cache, the register, etc., and was realized by performing.
[0003] In addition, the technique indicated by JP,3-250225,A is mentioned as this related kind of a conventional technique.
[0004]
[Problem(s) to be Solved by the Invention] However, in the above conventional techniques, by the operating environment of an actual information processor, two or more jobs also including user application jobs including an operating system are performed by asynchronous, under such a situation, the contents, such as memory in an information processor, a cache, a register, and TLB, are seen from an object program, and random data are arranged.
[0005] In this case, it is known that random data are arranged on the resource in an information processor like an operating environment, will cause malfunction when poor logic is inherent in the information processor under the situation that two or more conditions overlap, and poor logic will tend to be revealed from a condition with an information processor similar to the condition after power on reset or it.
[0006] Therefore, in the logic verification technique of the conventional technique, since logic verification by the information-processor milieu interne equivalent to a real operating environment was not carried out, there was a problem that verification precision was low.
[0007] Then, the purpose of this invention builds an environment equivalent to an actual user program being performed under operation system, and is by enabling logic verification under an environment equivalent to this real operating environment to offer the logic verification environment control unit which can realize highly precise logic verification.
[0008] Moreover, other purposes of this invention are to offer the logic verification environment control unit which can perform back trace from the interruption address of this failure generating, and can realize failure analysis easily by this trace information at the time of failure generating.
[0009] The other purposes and the new description will become clear from description and the accompanying drawing of this specification along [said] this invention.
[0010]
[Means for Solving the Problem] It will be as follows if the outline of a typical thing is briefly explained among invention indicated in this application.
[0011] Namely, the logic verification environment control unit of this invention The random-number data generator which is a logic verification environment control unit to the logic simulation equipment which carries out circuit simulation of gate level, or an information processor, and creates random random-number data, The random-number data setting device which sets this random-number data to the data storage means in logic simulation equipment or an information processor, for example, memory, a cache, TLB, a register, etc., The interrupt control equipment which

controls the interruption generated within logic simulation equipment or an information processor, It has the environment condition control unit which carries out control of operator interface control, logic simulation equipment, or an information processor and the edit output of trace information.
[0012] In this case, in case the entry data of said data storage means are created, it creates so that this entry data, for example, a parity bit, Hamming code (ECC:Error Correcting Code), etc. may be adjusted in the logical format of logic simulation equipment or an information processor, and random-number data are embedded to entry data.
[0013] Moreover, it is made for said random-number data generator to generate the instruction which operates on specific logic simulation equipment or an information processor and address data, for example, an absolute-instruction code, operand address data, etc.
[0014] Furthermore, in case said random-number data generator creates the random-number data set as a data storage means, it is made to generate an interruption generating instruction operand code in dozens to hundreds of byte pitch.
[0015] Moreover, said interrupt-processing equipment checks the interruption generating address and an interrupt type, carries out back trace in the range traceable from the interruption generating address, and saves trace information.
[0016]
[Function] According to the above mentioned logic verification environment control unit, by having a random-number data generator, a random-number data setting device, interrupt control equipment, and an environment condition control unit For example, it creates so that entry data, such as a parity bit and Hamming code, may be adjusted in a logical format. The random-number data based on a random-number data generator are embedded to this entry data. This entry data can be set as a data storage means with a random-number data setting device, and the resource in logic simulation equipment or an information processor can be set as an environment equivalent to a real operating environment before logic verification program execution.
[0017] Moreover, it can be made to operate only on specific logic simulation equipment or an information processor according to generating of a specific absolute-instruction code, operand address data, etc.
[0018] Furthermore, by an interruption generating instruction operand code's being generated by dozens to hundreds of byte pitch, and checking the interruption generating address and an interrupt type, back trace can be carried out in the range traceable from the interruption generating address, and this trace information can be saved.
[0019] An equipment environment equivalent to an actual user program being performed under operation system by this is built, and while realizing highly precise logic verification by enabling logic verification under this environment, failure analysis is easily realizable at the time of failure generating.
[0020]
[Example] The explanatory view showing processing of the block diagram showing the logic verification environment control unit whose drawing 1 is one example of this invention, a random-number data generator [in / in drawing 2 / the logic verification environment control unit of this example], and a random-number data setting device, the flow chart Fig. in which drawing 3 shows cache data-origination processing of a random-number data generator in this example, the flow chart Fig. in which drawing 4 shows data origination / setting processing, and drawing 5 are the flow chart Figs. showing logic verification processing of having used the logic verification environment control unit of this example. [the hardware register of a random-number data generator and a random-number data setting device]
[0021] First, drawing 1 explains the configuration of the logic verification environment control unit of this example.
[0022] It considers as the logic verification environment control unit to an information processor, and the logic verification environment control unit of this example consists of trace information files 5 for storing the model-information file 4 and back trace information that the verification program file 3 in which the logic verification program of the target information processor 1, a system console 2, and execute form is stored, the size of each RAM, an entry, etc. are stored etc.
[0023] Random-number data are embedded for the object RAM of the environment condition control device 11 and an information processor which carries out operator interface control, control of each equipment, and edit output processing of trace information in the target information processor 1. The logic verification program performed under the random-number data generator 12 which creates an entry, the random-number data setting device 13 which sets the created entry as Object RAM, and the set-up environment overruns recklessly by poor logic. When an interrupt occurs with an interruption generating instruction, it has memory 15, a cache 16, a register 17, TLB18, etc. as an object RAM of interrupt control equipment 14 and a data storage means which carries out the recovery and back trace.
[0024] Next, an operation of this example is explained.
[0025] Introduction and drawing 2 explain processing of the random-number data generator 12 and the random-number data setting device 13.

[0026] If the random-number data generator 12 is started from the environment condition control device 11, it will first read a format of sizes, such as the memory 15 of each RAM in an information processor 1, a cache 16, a register 17, and TLB18, and an entry, a block/linesize, etc. from the model-information file 4, and will generate the entry data of each RAM according to the directions.
[0027] Under the present circumstances, the generate time of entry data which has parity bits, Hamming codes (ECC), etc., such as TLB18 and a cache address array, generates the data suitable for the logical format of an information processor 1 to the predetermined bit position according to directions of the model-information file 4.
[0028] Moreover, the data of memory 15 and a cache 16 generate random-number data per the specified block/linesize. In this case, the machine language code of the interrupt instruction of dedication which makes interrupt control equipment 14 generate the interruption for moving control is embedded at several bytes of the last of each block/Rhine. And the random-number data generator 12 moves control to the random-number data setting device 13 after generating entry data.
[0029] Furthermore, the random-number data setting device 13 sets up the entry data for the size (the number of entries) registered into the model-information file 4 by predetermined RAM in the entry data generated with the random-number data generator 12.
[0030] Next, the flow of drawing 3 explains creation processing of the code cache data of the random-number data generator 12.
[0031] The random-number data generator 12 creates a Hamming code (ECC) part for code cache entry information according to read-out (step 301) and this entry information from the model-information file 4 first (step 302), carries out random number generation of the code data actual next, and creates random-number data (step 303).
[0032] Furthermore, interruption generating instruction code is embedded at the last of a block of a code cache entry (step 304), and the random-number data setting device 13 is started (step 305).
[0033] Next, the flow of drawing 4 explains data origination / setting processing of a hardware (H/W) register.
[0034] When waiting arises in advanced-control processing of an information processor 1 (i.e., when the next MVC (data transfer from memory to memory) instruction is using the contents of the register decided with LD (data transfer from memory to register) instruction as a memory address), since address computation processing of an MVC instruction cannot be performed until processing of this LD instruction is completed, waiting will arise in processing and a register will once be in idle status.
[0035] When such a condition occurs, the environment condition control device 11 senses that the bubble occurred with a lock for example, with A stage register (step 401), and issues generation directions of data to the random-number data generator 12.
[0036] Furthermore, the random-number data generator 12 creates the data of A stage register (step 402), and takes out the setup instruction of data to the random-number data setting device 13.
[0037] And the random-number data setting device 13 embeds the data based on the random-number value generated by A stage register (step 403).
[0038] Next, drawing 5 explains the processing flow of the logic verification using the logic verification environment control unit of this example.
[0039] First, the environment condition control device 11 starts the random-number data generator 12 with directions of an operator (step 501). And the random-number data generator 12 reads the creation information of the entry data of each RAM from the model-information file 4 (step 502), and creates the entry data of each RAM according to this model information (step 503).
[0040] Furthermore, the random-number data generator 12 starts the random-number data setting device 13 (step 504), and the created entry data are initialized to each RAM according to the size of each RAM by which the random-number data setting device 13 is registered into the model-information file 4 (step 505).
[0041] And control is returned to the environment condition control unit 11 after initialization termination (step 506), and this environment condition control unit 11 loads the logic verification program which corresponds with directions of an operator to memory 15 (step 507), and starts this logic verification program (step 508).
[0042] Furthermore, if logic verification program execution is completed normally, the processing whose environment condition control unit 11 carries out the next processing demand to an operator will be repeated.
[0043] When a logic verification program overruns recklessly (step 509), with an interruption generating instruction, control moves to interrupt control equipment 14 (step 510), goes back to the address in which back trace is possible, collects trace information, and registers with the trace information file 5 (step 511).
[0044] And control edits trace information into the environment condition control unit 11, return (step 512) and the

environment condition control unit 11 output this trace information to a system console 2 (step 513), and analysis of the cause of an overrun is performed by this.

[0045] Therefore, according to the logic verification environment control unit of this example, it has the environment condition control device 11, the random-number data generator 12, the random-number data setting device 13, and interrupt control equipment 14. By embedding random-number data to entry data, and initializing this entry data to memory 15, a cache 16, a register 17, TLB18, etc. of each RAM in an information processor 1 The data storage means in the information news processor 1 set as the object of logic verification can be set as an environment equal to an actual user program being performed, and the logic verification under a real operating environment is attained by this.

[0046] Moreover, since back trace becomes possible from the interruption generating address with an interruption generating instruction also when a logic verification program overruns recklessly, this trace information can perform easily failure analysis at the time of failure generating.

[0047] As mentioned above, although invention made by this invention person was concretely explained based on the example, it cannot be overemphasized that it can change variously in the range which this invention is not limited to said example and does not deviate from the summary.

[0048] For example, although the case where the object of the logic verification about the logic verification environment control unit of this example was an information processor 1 was explained, this invention is not limited to said example and can be applied widely also about the logic simulation equipment which carries out circuit simulation of gate level.

[0049] Moreover, you may make it generate an absolute-instruction code, operand address data, etc. which operate, for example on a specific information processor as a random-number data generator 12.

[0050]

[Effect of the Invention] It will be as follows if the effectiveness acquired by the typical thing among invention indicated in this application is explained briefly.

[0051] (1) Since it can create so that entry data may be adjusted in a logical format by having . random-number data generator, a random-number data setting device, interrupt control equipment, and an environment condition control device, and random-number data can be embedded to this entry data and it can be set as a data storage means, the resource in logic simulation equipment or an information processor can be built by the environment equivalent to a real operating environment before logic verification program execution.

[0052] (2) It can be made to operate only on specific logic simulation equipment or an information processor, when . random-number data generator generates a specific instruction and address data.

[0053] (3) Since . random-number data generator can generate an interruption generating instruction operand code in dozens to hundreds of byte pitch, interrupt-processing equipment can carry out back trace from the interruption generating address of this interruption generating instruction operand code and trace information can be saved, failure analysis when a failure occurs can be performed easily.

[0054] (4) . above (1) - (3) While realizing highly precise logic verification by enabling logic verification under a real operating environment to the logic simulation equipment or the information processor used as the candidate for logic verification, the logic verification environment control unit with which failure analysis at the time of failure generating is enabled easily can be obtained.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Industrial Application] About the logic verification technique of the logical circuit which constitutes logic simulation equipment or an information processor, especially, this invention is complicated, is applied to the suitable logic verification environment control unit for highly precise verification of the big logical circuit of a scale, and relates to an effective technique.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] For example, by the conventional logic verification method, from the condition similar to the condition that the memory in an information processor after power on reset, a cache, a register, TLB (Table Look-aside Buffer), etc. were initialized (zero clear), or it, the information processor set the logic verification program to a memory load or the memory which corresponds verification program data, the cache, the register, etc., and was realized by performing.
[0003] In addition, the technique indicated by JP,3-250225,A is mentioned as this related kind of a conventional technique.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] It will be as follows if the effectiveness acquired by the typical thing among invention indicated in this application is explained briefly.
[0051] (1) Since it can create so that entry data may be adjusted in a logical format by having . random-number data generator, a random-number data setting device, interrupt control equipment, and an environment condition control device, and random-number data can be embedded to this entry data and it can be set as a data storage means, the resource in logic simulation equipment or an information processor can be built by the environment equivalent to a real operating environment before logic verification program execution.
[0052] (2) It can be made to operate only on specific logic simulation equipment or an information processor, when . random-number data generator generates a specific instruction and address data.
[0053] (3) Since . random-number data generator can generate an interruption generating instruction operand code in dozens to hundreds of byte pitch, interrupt-processing equipment can carry out back trace from the interruption generating address of this interruption generating instruction operand code and trace information can be saved, failure analysis when a failure occurs can be performed easily.
[0054] (4) . above (1) - (3) While realizing highly precise logic verification by enabling logic verification under a real operating environment to the logic simulation equipment or the information processor used as the candidate for logic verification, the logic verification environment control unit with which failure analysis at the time of failure generating is enabled easily can be obtained.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] However, in the above conventional techniques, by the operating environment of an actual information processor, two or more jobs also including user application jobs including an operating system are performed by asynchronous, under such a situation, the contents, such as memory in an information processor, a cache, a register, and TLB, are seen from an object program, and random data are arranged.
[0005] In this case, it is known that random data are arranged on the resource in an information processor like an operating environment, will cause malfunction when poor logic is inherent in the information processor under the situation that two or more conditions overlap, and poor logic will tend to be revealed from a condition with an information processor similar to the condition after power on reset or it.
[0006] Therefore, in the logic verification technique of the conventional technique, since logic verification by the information-processor milieu interne equivalent to a real operating environment was not carried out, there was a problem that verification precision was low.
[0007] Then, the purpose of this invention builds an environment equivalent to an actual user program being performed under operation system, and is by enabling logic verification under an environment equivalent to this real operating environment to offer the logic verification environment control unit which can realize highly precise logic verification.
[0008] Moreover, other purposes of this invention are to offer the logic verification environment control unit which can perform back trace from the interruption address of this failure generating, and can realize failure analysis easily by this trace information at the time of failure generating.
[0009] The other purposes and the new description will become clear from description and the accompanying drawing of this specification along [said] this invention.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] It will be as follows if the outline of a typical thing is briefly explained among invention indicated in this application.
[0011] Namely, the logic verification environment control unit of this invention The random-number data generator which is a logic verification environment control unit to the logic simulation equipment which carries out circuit simulation of gate level, or an information processor, and creates random random-number data, The random-number data setting device which sets this random-number data to the data storage means in logic simulation equipment or an information processor, for example, memory, a cache, TLB, a register, etc., The interrupt control equipment which controls the interruption generated within logic simulation equipment or an information processor, It has the environment condition control unit which carries out control of operator interface control, logic simulation equipment, or an information processor and the edit output of trace information.
[0012] In this case, in case the entry data of said data storage means are created, it creates so that this entry data, for example, a parity bit, Hamming code (ECC:Error Correcting Code), etc. may be adjusted in the logical format of logic simulation equipment or an information processor, and random-number data are embedded to entry data.
[0013] Moreover, it is made for said random-number data generator to generate the instruction which operates on specific logic simulation equipment or an information processor and address data, for example, an absolute-instruction code, operand address data, etc.
[0014] Furthermore, in case said random-number data generator creates the random-number data set as a data storage means, it is made to generate an interruption generating instruction operand code in dozens to hundreds of byte pitch.
[0015] Moreover, said interrupt-processing equipment checks the interruption generating address and an interrupt type, carries out back trace in the range traceable from the interruption generating address, and saves trace information.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

OPERATION

[Function] According to the above mentioned logic verification environment control unit, by having a random-number data generator, a random-number data setting device, interrupt control equipment, and an environment condition control unit For example, it creates so that entry data, such as a parity bit and Hamming code, may be adjusted in a logical format. The random-number data based on a random-number data generator are embedded to this entry data. This entry data can be set as a data storage means with a random-number data setting device, and the resource in logic simulation equipment or an information processor can be set as an environment equivalent to a real operating environment before logic verification program execution.
[0017] Moreover, it can be made to operate only on specific logic simulation equipment or an information processor according to generating of a specific absolute-instruction code, operand address data, etc.
[0018] Furthermore, by an interruption generating instruction operand code's being generated by dozens to hundreds of byte pitch, and checking the interruption generating address and an interrupt type, back trace can be carried out in the range traceable from the interruption generating address, and this trace information can be saved.
[0019] An equipment environment equivalent to an actual user program being performed under operation system by this is built, and while realizing highly precise logic verification by enabling logic verification under this environment, failure analysis is easily realizable at the time of failure generating.

[Translation done.]

Family list
4 family members for:
EP0768612
Derived from 3 applications.

Back to EP0768612

1 **Method and apparatus for generating structured document**
Publication info: **EP0768612 A2** - 1997-04-16
EP0768612 A3 - 2005-02-16

2 **METHOD AND DEVICE FOR GENERATING STRUCTURED DOCUMENT**
Publication info: **JP9069101 A** - 1997-03-11

3 **Method and apparatus for generating structured document**
Publication info: **US6014680 A** - 2000-01-11

Data supplied from the ***esp@cenet*** database - Worldwide

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

EXAMPLE

[Example] The explanatory view showing processing of the block diagram showing the logic verification environment control unit whose drawing 1 is one example of this invention, a random-number data generator [in / in drawing 2 / the logic verification environment control unit of this example], and a random-number data setting device, the flow chart Fig. in which drawing 3 shows cache data-origination processing of a random-number data generator in this example, the flow chart Fig. in which drawing 4 shows data origination / setting processing, and drawing 5 are the flow chart Figs. showing logic verification processing of having used the logic verification environment control unit of this example. [the hardware register of a random-number data generator and a random-number data setting device]
[0021] First, drawing 1 explains the configuration of the logic verification environment control unit of this example.
[0022] It considers as the logic verification environment control unit to an information processor, and the logic verification environment control unit of this example consists of trace information files 5 for storing the model-information file 4 and back trace information that the verification program file 3 in which the logic verification program of the target information processor 1, a system console 2, and execute form is stored, the size of each RAM, an entry, etc. are stored etc.
[0023] Random-number data are embedded for the object RAM of the environment condition control device 11 and an information processor which carries out operator interface control, control of each equipment, and edit output processing of trace information in the target information processor 1. The logic verification program performed under the random-number data generator 12 which creates an entry, the random-number data setting device 13 which sets the created entry as Object RAM, and the set-up environment overruns recklessly by poor logic. When an interrupt occurs with an interruption generating instruction, it has memory 15, a cache 16, a register 17, TLB18, etc. as an object RAM of interrupt control equipment 14 and a data storage means which carries out the recovery and back trace.
[0024] Next, an operation of this example is explained.
[0025] Introduction and drawing 2 explain processing of the random-number data generator 12 and the random-number data setting device 13.
[0026] If the random-number data generator 12 is started from the environment condition control device 11, it will first read a format of sizes, such as the memory 15 of each RAM in an information processor 1, a cache 16, a register 17, and TLB18, and an entry, a block/linesize, etc. from the model-information file 4, and will generate the entry data of each RAM according to the directions.
[0027] Under the present circumstances, the generate time of entry data which has parity bits, Hamming codes (ECC), etc., such as TLB18 and a cache address array, generates the data suitable for the logical format of an information processor 1 to the predetermined bit position according to directions of the model-information file 4.
[0028] Moreover, the data of memory 15 and a cache 16 generate random-number data per the specified block/linesize. In this case, the machine language code of the interrupt instruction of dedication which makes interrupt control equipment 14 generate the interruption for moving control is embedded at several bytes of the last of each block/Rhine. And the random-number data generator 12 moves control to the random-number data setting device 13 after generating entry data.
[0029] Furthermore, the random-number data setting device 13 sets up the entry data for the size (the number of entries) registered into the model-information file 4 by predetermined RAM in the entry data generated with the random-number data generator 12.
[0030] Next, the flow of drawing 3 explains creation processing of the code cache data of the random-number data generator 12.
[0031] The random-number data generator 12 creates a Hamming code (ECC) part for code cache entry information according to read-out (step 301) and this entry information from the model-information file 4 first (step 302), carries

out random number generation of the code data actual next, and creates random-number data (step 303).
[0032] Furthermore, interruption generating instruction code is embedded at the last of a block of a code cache entry (step 304), and the random-number data setting device 13 is started (step 305).
[0033] Next, the flow of drawing 4 explains data origination / setting processing of a hardware (H/W) register.
[0034] When waiting arises in advanced-control processing of an information processor 1 (i.e., when the next MVC (data transfer from memory to memory) instruction is using the contents of the register decided with LD (data transfer from memory to register) instruction as a memory address), since address computation processing of an MVC instruction cannot be performed until processing of this LD instruction is completed, waiting will arise in processing and a register will once be in idle status.
[0035] When such a condition occurs, the environment condition control device 11 senses that the bubble occurred with a lock for example, with A stage register (step 401), and issues generation directions of data to the random-number data generator 12.
[0036] Furthermore, the random-number data generator 12 creates the data of A stage register (step 402), and takes out the setup instruction of data to the random-number data setting device 13.
[0037] And the random-number data setting device 13 embeds the data based on the random-number value generated by A stage register (step 403).
[0038] Next, drawing 5 explains the processing flow of the logic verification using the logic verification environment control unit of this example.
[0039] First, the environment condition control device 11 starts the random-number data generator 12 with directions of an operator (step 501). And the random-number data generator 12 reads the creation information of the entry data of each RAM from the model-information file 4 (step 502), and creates the entry data of each RAM according to this model information (step 503).
[0040] Furthermore, the random-number data generator 12 starts the random-number data setting device 13 (step 504), and the created entry data are initialized to each RAM according to the size of each RAM by which the random-number data setting device 13 is registered into the model-information file 4 (step 505).
[0041] And control is returned to the environment condition control unit 11 after initialization termination (step 506), and this environment condition control unit 11 loads the logic verification program which corresponds with directions of an operator to memory 15 (step 507), and starts this logic verification program (step 508).
[0042] Furthermore, if logic verification program execution is completed normally, the processing whose environment condition control unit 11 carries out the next processing demand to an operator will be repeated.
[0043] When a logic verification program overruns recklessly (step 509), with an interruption generating instruction, control moves to interrupt control equipment 14 (step 510), goes back to the address in which back trace is possible, collects trace information, and registers with the trace information file 5 (step 511).
[0044] And control edits trace information into the environment condition control unit 11, return (step 512) and the environment condition control unit 11 output this trace information to a system console 2 (step 513), and analysis of the cause of an overrun is performed by this.
[0045] Therefore, according to the logic verification environment control unit of this example, it has the environment condition control device 11, the random-number data generator 12, the random-number data setting device 13, and interrupt control equipment 14. By embedding random-number data to entry data, and initializing this entry data to memory 15, a cache 16, a register 17, TLB18, etc. of each RAM in an information processor 1 The data storage means in the information news processor 1 set as the object of logic verification can be set as an environment equal to an actual user program being performed, and the logic verification under a real operating environment is attained by this.
[0046] Moreover, since back trace becomes possible from the interruption generating address with an interruption generating instruction also when a logic verification program overruns recklessly, this trace information can perform easily failure analysis at the time of failure generating.
[0047] As mentioned above, although invention made by this invention person was concretely explained based on the example, it cannot be overemphasized that it can change variously in the range which this invention is not limited to said example and does not deviate from the summary.
[0048] For example, although the case where the object of the logic verification about the logic verification environment control unit of this example was an information processor 1 was explained, this invention is not limited to said example and can be applied widely also about the logic simulation equipment which carries out circuit simulation of gate level.
[0049] Moreover, you may make it generate an absolute-instruction code, operand address data, etc. which operate, for

example on a specific information processor as a random-number data generator 12.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] It is the block diagram showing the logic verification environment control unit which is one example of this invention.
[Drawing 2] It is the explanatory view showing processing of the random-number data generator in the logic verification environment control unit of this example and a random-number data setting device.
[Drawing 3] In this example, it is the flow chart Fig. showing cache data origination processing of a random-number data generator.
[Drawing 4] In this example, it is the flow chart Fig. showing data origination / setting processing of the hardware register of a random-number data generator and a random-number data setting device.
[Drawing 5] It is the flow chart Fig. showing the logic verification processing using the logic verification environment control unit of this example.
[Description of Notations]
1 Information Processor
2 System Console
3 Verification Program File
4 Model-Information File
5 Trace Information File
11 Environment Condition Control Unit
12 Random-Number Data Generator
13 Random-Number Data Setting Device
14 Interrupt Control Equipment
15 Memory
16 Cache
17 Register
18 TLB

[Translation done.]

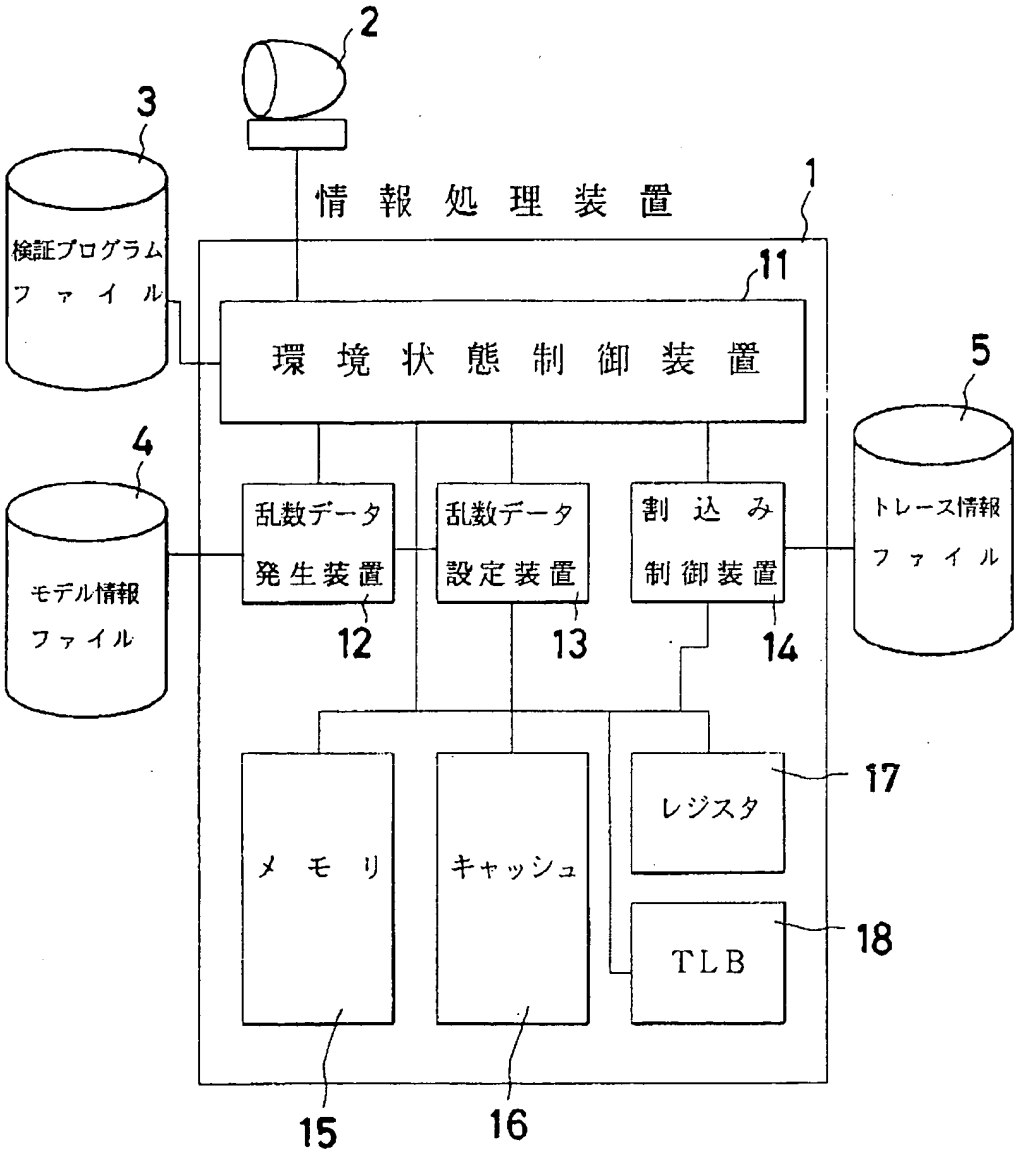
* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

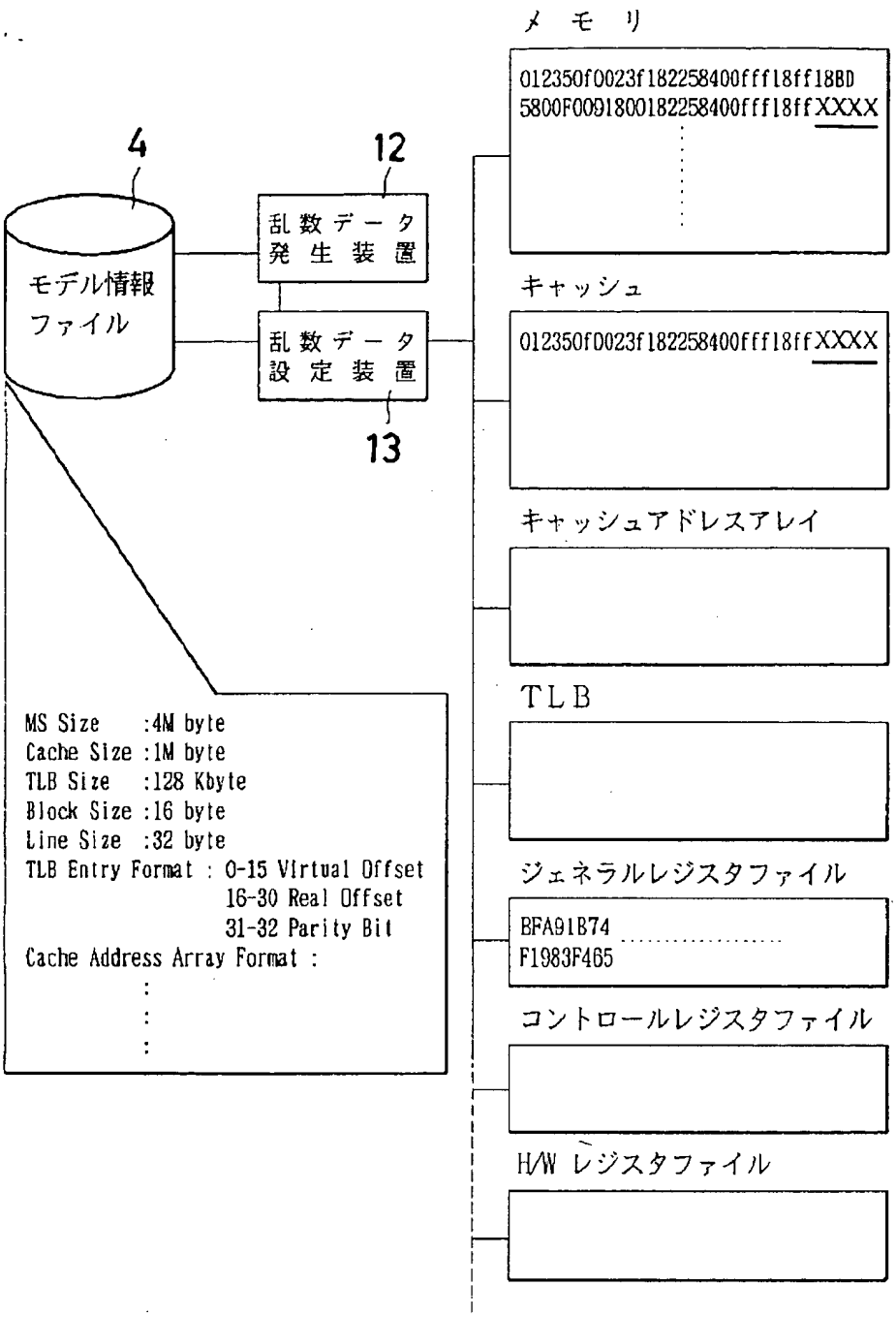
[Drawing 1]

図 1



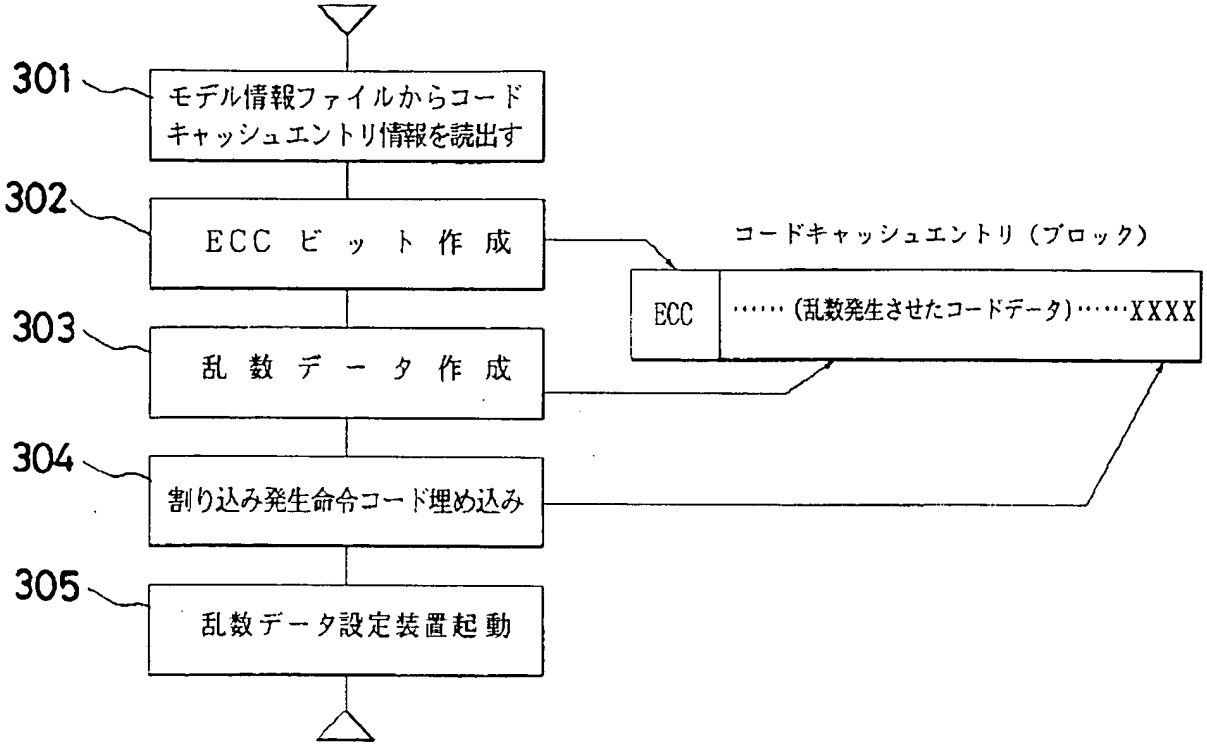
[Drawing 2]

図 2


メモリ
012350f0023f182258400fff18ff18BD
5800F00918001822584OOfff18ffXXXX
4
12
乱数データ発生装置
モデル情報ファイル
乱数データ設定装置
13
キャッシュ
012350f0023f182258400fff18ffXXXX
キャッシュアドレスアレイ
TLB
ジェネラルレジスタファイル
BFA91B74
F1983F465
コントロールレジスタファイル
H/W レジスタファイル
MS Size :4M byte
Cache Size :1M byte
TLB Size :128 Kbyte
Block Size :16 byte
Line Size :32 byte
TLB Entry Format : 0-15 Virtual Offset
16-30 Real Offset
31-32 Parity Bit
Cache Address Array Format :

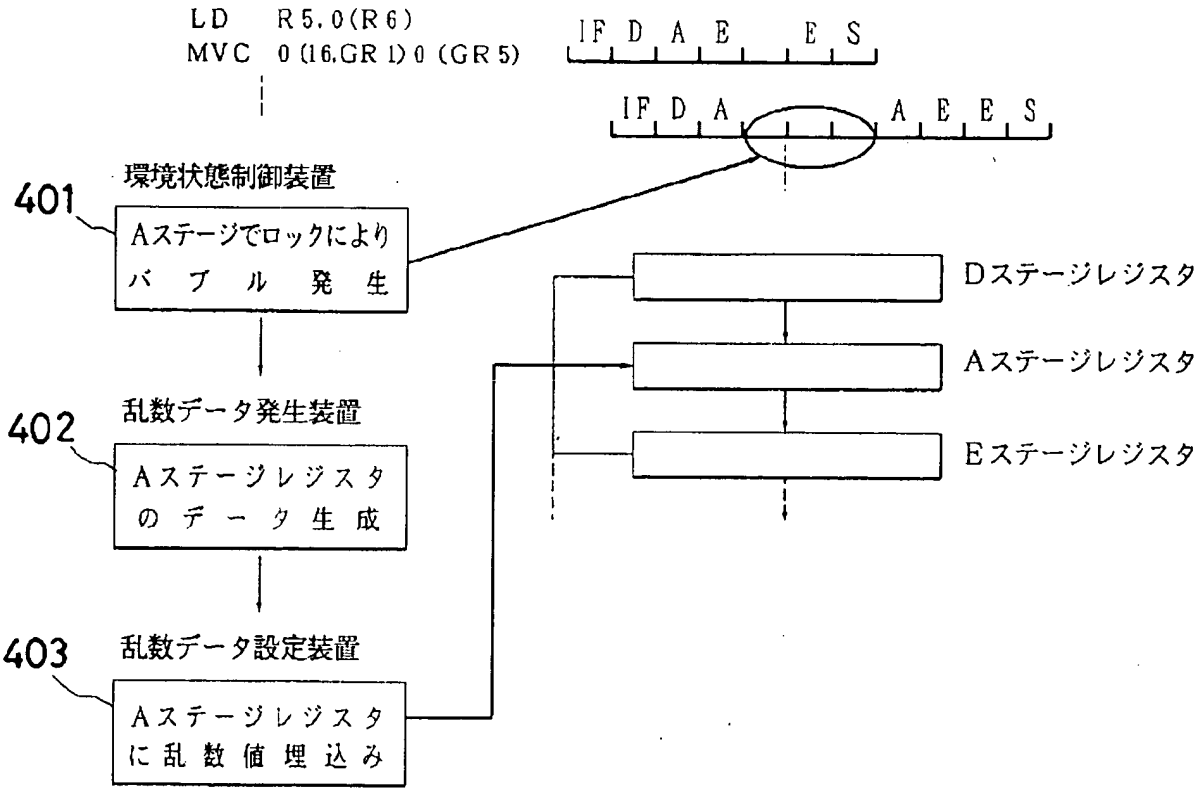
[Drawing 3]

図 3



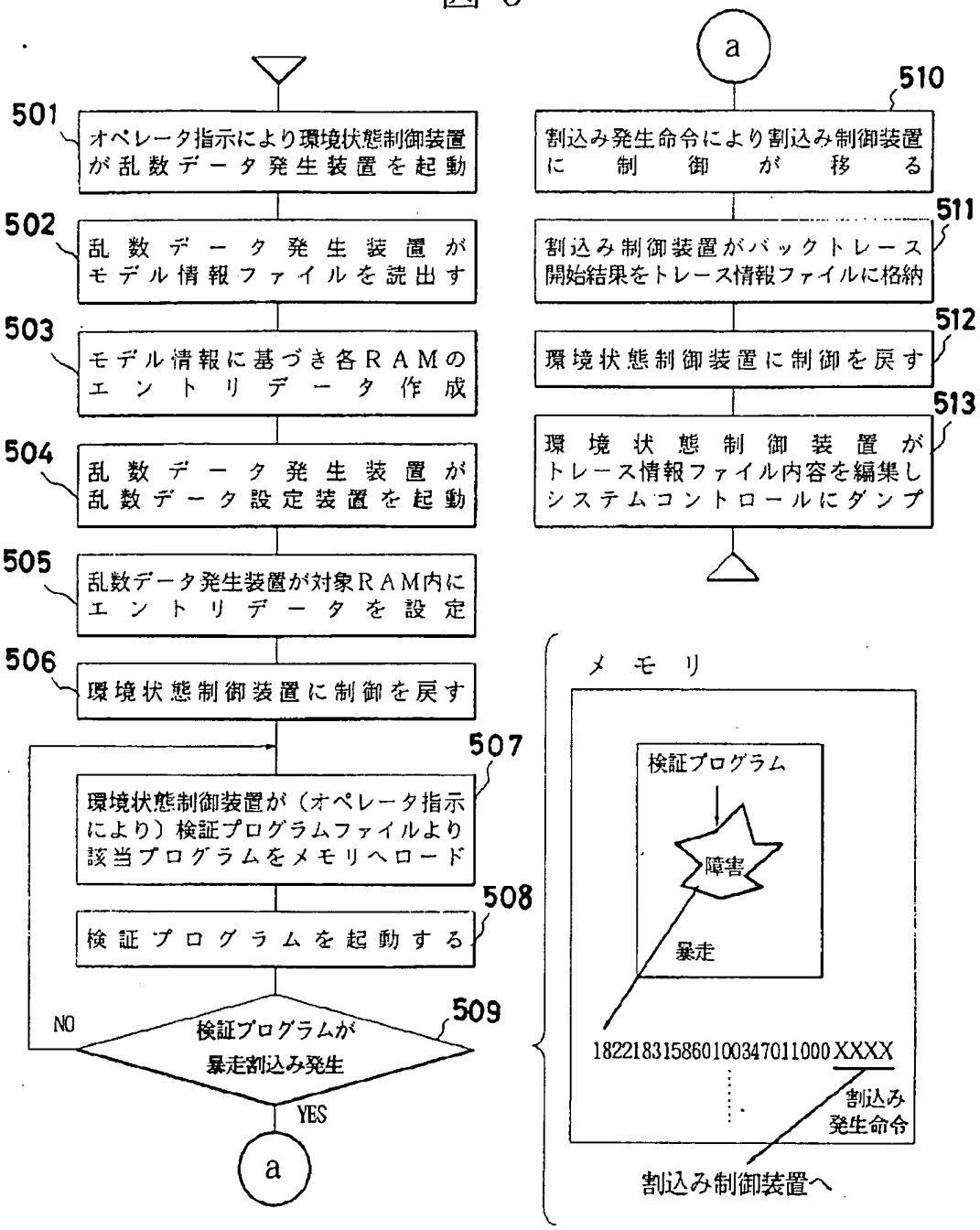
[Drawing 4]

図 4



[Drawing 5]

図 5



[Translation done.]

09/725,787

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平5－274385

(43)公開日　平成５年(1993)10月22日

(51)Int.Cl.[5]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	15/60	３６０　Ｄ	7922－5L		
	9/46	３１０　Ｋ	8120－5B		
	11/22	３１０　Ｃ	8323－5B		
	11/26	３１０	8323－5B		

審査請求　未請求　請求項の数５(全 10 頁)

(21)出願番号　特願平4－66527

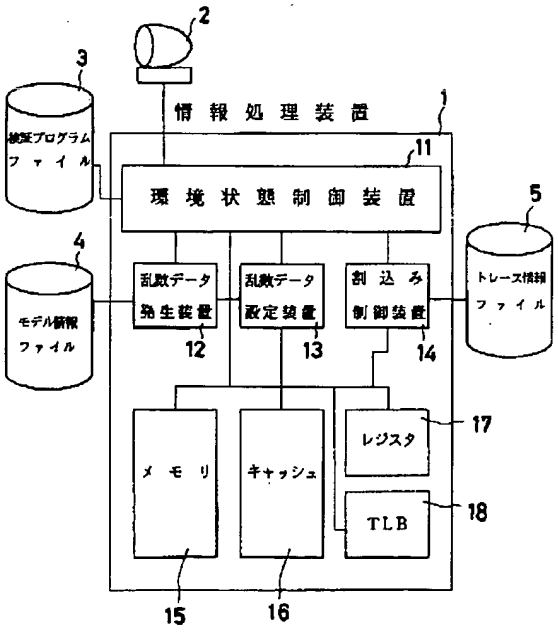
(22)出願日　平成４年(1992)３月25日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　鈴木　薫
神奈川県秦野市堀山下１番地　株式会社日立製作所神奈川工場内
(74)代理人　弁理士　筒井　大和

(54)【発明の名称】　論理検証環境制御装置

(57)【要約】
【目的】　論理シミュレーション装置または情報処理装置に対する論理検証の際、実使用環境と等価な環境設定によって高精度な論理検証を実現すると共に、障害発生時の障害解析を容易に実現できる論理検証環境制御装置を提供する。
【構成】　論理検証の対象となる情報処理装置１、システムコンソール２、検証プログラムファイル３、モデル情報ファイル４、トレース情報ファイル５などから構成され、情報処理装置１には、環境状態制御装置１１、乱数データを埋め込んだエントリの乱数データ発生装置１２、エントリの乱数データ設定装置１３、プログラムの暴走時にバックトレースを実施する割込み制御装置１４、メモリ１５、キャッシュ１６、レジスタ１７およびＴＬＢ１８が備えられている。そして、論理検証プログラムの実行以前に情報処理装置１内資源が初期設定される。

図１



1

【特許請求の範囲】

【請求項1】　ゲートレベルの回路シミュレーションを実施する論理シミュレーション装置、または情報処理装置に対する論理検証環境制御装置であって、ランダムな乱数データを作成する乱数データ発生装置と、該乱数データを前記論理シミュレーション装置または情報処理装置内のデータ格納手段に設定する乱数データ設定装置と、前記論理シミュレーション装置または情報処理装置内で発生した割込みを制御する割込み制御装置と、オペレータインタフェース制御、前記論理シミュレーション装置または情報処理装置の制御、およびトレース情報の編集出力を実施する環境状態制御装置とを備え、論理検証プログラムの実行以前に、前記論理シミュレーション装置または情報処理装置内資源を実使用環境と等価な環境に設定することを特徴とする論理検証環境制御装置。

【請求項2】　前記データ格納手段のエントリデータを作成する際、該エントリデータを前記論理シミュレーション装置または情報処理装置の論理フォーマットに整合するように作成し、該エントリデータに前記乱数データを埋め込むことを特徴とする請求項1記載の論理検証環境制御装置。

【請求項3】　前記乱数データ発生装置が、特定の前記論理シミュレーション装置または情報処理装置上で動作する命令およびアドレスデータを発生することを特徴とする請求項1記載の論理検証環境制御装置。

【請求項4】　前記乱数データ発生装置が、前記データ格納手段に設定する乱数データを作成する際、数十から数百バイトピッチに割込み発生命令オペランドコードを発生することを特徴とする請求項1記載の論理検証環境制御装置。

【請求項5】　前記割込み処理装置が、割込み発生アドレスおよび割込み種別をチェックし、割込み発生アドレスからトレース可能な範囲でバックトレースを実施してトレース情報をセーブすることを特徴とする請求項1記載の論理検証環境制御装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、論理シミュレーション装置または情報処理装置などを構成する論理回路の論理検証技術に関し、特に複雑で規模の大きな論理回路の高精度な検証に好適な論理検証環境制御装置に適用して有効な技術に関する。

【0002】

【従来の技術】たとえば、従来の論理検証方式では、情報処理装置がパワーオンリセット後の情報処理装置内メモリ、キャッシュ、レジスタ、ＴＬＢ（Table Look-aside Buffer ）などがイニシャライズ（ゼロクリア）された状態、またはそれと類似な状態から論理検証プログラムをメモリロード、または検証プログラムデータを該当するメモリ、キャッシュ、レジスタなどに設定し、実行

2

することで実現されていた。

【0003】なお、関連するこの種の従来技術として、たとえば特開平3－250225号公報に記載される技術などが挙げられる。

【0004】

【発明が解決しようとする課題】ところが、前記のような従来技術において、実際の情報処理装置の使用環境では、オペレーティングシステムを始めとして、ユーザアプリケーションジョブも含めて複数のジョブが非同期に実行されており、このような状況下では、情報処理装置内メモリ、キャッシュ、レジスタ、ＴＬＢなどの内容は対象プログラムから見てランダムなデータが配されている。

【0005】この場合に、情報処理装置がパワーオンリセット後の状態、またはそれと類似な状態より、使用環境のように情報処理装置内資源にランダムなデータが配され、複数の条件が重なり合う状況下では、もし情報処理装置に論理不良が内在している場合に誤動作を起こして論理不良が発覚し易いことが知られている。

【0006】従って、従来技術の論理検証技術においては、実使用環境と等価な情報処理装置内環境での論理検証が実施されていないために、検証精度が低いという問題があった。

【0007】そこで、本発明の目的は、オペレーションシステム下で実際のユーザプログラムが実行されるのと等価な環境を構築し、この実使用環境と等価な環境下での論理検証を可能とすることによって高精度な論理検証を実現することができる論理検証環境制御装置を提供することにある。

【0008】また、本発明の他の目的は、障害発生時に、この障害発生の割込みアドレスからのバックトレースを行い、このトレース情報によって障害解析を容易に実現することができる論理検証環境制御装置を提供することにある。

【0009】本発明の前記ならびにその他の目的と新規な特徴は、本明細書の記述および添付図面から明らかになるであろう。

【0010】

【課題を解決するための手段】本願において開示される発明のうち、代表的なものの概要を簡単に説明すれば、下記のとおりである。

【0011】すなわち、本発明の論理検証環境制御装置は、ゲートレベルの回路シミュレーションを実施する論理シミュレーション装置、または情報処理装置に対する論理検証環境制御装置であって、ランダムな乱数データを作成する乱数データ発生装置と、この乱数データを論理シミュレーション装置または情報処理装置内のデータ格納手段、たとえばメモリ、キャッシュ、ＴＬＢ、レジスタなどに設定する乱数データ設定装置と、論理シミュレーション装置または情報処理装置内で発生した割込み

を制御する割込み制御装置と、オペレータインタフェース制御、論理シミュレーション装置または情報処理装置の制御、およびトレース情報の編集出力を実施する環境状態制御装置とを備えるものである。

【００１２】この場合に、前記データ格納手段のエントリデータを作成する際、このエントリデータ、たとえばパリティビット、ハミングコード（ＥＣＣ：Error Correcting Code ）などを論理シミュレーション装置または情報処理装置の論理フォーマットに整合するように作成し、エントリデータに乱数データを埋め込むようにしたものである。

【００１３】また、前記乱数データ発生装置が、特定の論理シミュレーション装置または情報処理装置上で動作する命令およびアドレスデータ、たとえば機械語命令コード、オペランドアドレスデータなどを発生するようにしたものである。

【００１４】さらに、前記乱数データ発生装置が、データ格納手段に設定する乱数データを作成する際、数十から数百バイトピッチに割込み発生命令オペランドコードを発生するようにしたものである。

【００１５】また、前記割込み処理装置が、割込み発生アドレスおよび割込み種別をチェックし、割込み発生アドレスからトレース可能な範囲でバックトレースを実施してトレース情報をセーブするようにしたものである。

【００１６】

【作用】前記した論理検証環境制御装置によれば、乱数データ発生装置、乱数データ設定装置、割込み制御装置および環境状態制御装置が備えられることにより、たとえばパリティビットおよびハミングコードなどのエントリデータを論理フォーマットに整合するように作成し、このエントリデータに乱数データ発生装置による乱数データを埋め込み、このエントリデータを乱数データ設定装置によりデータ格納手段に設定し、論理検証プログラムの実行以前に、論理シミュレーション装置または情報処理装置内資源を実使用環境と等価な環境に設定することができる。

【００１７】また、特定の機械語命令コードおよびオペランドアドレスデータなどの発生により、特定の論理シミュレーション装置または情報処理装置上でのみ動作させることができる。

【００１８】さらに、数十から数百バイトピッチに割込み発生命令オペランドコードが発生され、割込み発生アドレスおよび割込み種別がチェックされることにより、割込み発生アドレスからトレース可能な範囲でバックトレースを実施し、このトレース情報をセーブすることができる。

【００１９】これにより、オペレーションシステム下で実際のユーザプログラムが実行されるのと等価な装置環境を構築し、この環境下での論理検証を可能とすることによって高精度な論理検証を実現すると共に、障害発生時に障害解析を容易に実現することができる。

【００２０】

【実施例】図１は本発明の一実施例である論理検証環境制御装置を示すブロック図、図２は本実施例の論理検証環境制御装置における乱数データ発生装置および乱数データ設定装置の処理を示す説明図、図３は本実施例において、乱数データ発生装置のキャッシュデータ作成処理を示すフローチャート図、図４は乱数データ発生装置および乱数データ設定装置のハードウェアレジスタのデータ作成／設定処理を示すフローチャート図、図５は本実施例の論理検証環境制御装置を用いた論理検証処理を示すフローチャート図である。

【００２１】まず、図１により本実施例の論理検証環境制御装置の構成を説明する。

【００２２】本実施例の論理検証環境制御装置は、たとえば情報処理装置に対する論理検証環境制御装置とされ、対象となる情報処理装置１、システムコンソール２、実行形式の論理検証プログラムが格納されている検証プログラムファイル３、各ＲＡＭのサイズおよびエントリなどが格納されているモデル情報ファイル４、バックトレース情報を格納するためのトレース情報ファイル５などから構成されている。

【００２３】対象となる情報処理装置１には、オペレータインタフェース制御、各装置の制御およびトレース情報の編集出力処理を実施する環境状態制御装置１１、情報処理装置の対象ＲＡＭに乱数データを埋め込み、エントリを作成する乱数データ発生装置１２、作成されたエントリを対象ＲＡＭに設定する乱数データ設定装置１３、設定された環境下で実行された論理検証プログラムが論理不良により暴走し、割込み発生命令で割込みが発生した場合、その回復処理およびバックトレースを実施する割込み制御装置１４、データ格納手段の対象ＲＡＭとしてのメモリ１５、キャッシュ１６、レジスタ１７およびＴＬＢ１８などが備えられている。

【００２４】次に、本実施例の作用について説明する。

【００２５】始めに、図２により乱数データ発生装置１２および乱数データ設定装置１３の処理を説明する。

【００２６】乱数データ発生装置１２は、環境状態制御装置１１から起動されると、まずモデル情報ファイル４から情報処理装置１内の各ＲＡＭのメモリ１５、キャッシュ１６、レジスタ１７、ＴＬＢ１８などのサイズ、エントリのフォーマット、ブロック／ラインサイズなどを読み出し、その指示に従って各ＲＡＭのエントリデータを生成する。

【００２７】この際、ＴＬＢ１８、キャッシュアドレスアレイなどのパリティビットやハミングコード（ＥＣＣ）などを有するエントリデータの生成時は、情報処理装置１の論理フォーマットに合ったデータをモデル情報ファイル４の指示に従い所定のビット位置に生成する。

【００２８】また、メモリ１５、キャッシュ１６のデー

タは指定されたブロック／ラインサイズ単位に乱数データを生成する。この場合、各ブロック／ラインの最後の数バイトに、割込み制御装置１４に制御を移すための割込みを発生させる専用の割込み命令の機械語コードを埋め込む。そして、エントリデータを生成後、乱数データ発生装置１２は乱数データ設定装置１３へ制御を移す。

【００２９】さらに、乱数データ設定装置１３は、乱数データ発生装置１２で生成されたエントリデータを所定のＲＡＭに、モデル情報ファイル４に登録されているサイズ（エントリ数）分のエントリデータを設定する。

【００３０】次に、図３のフローにより乱数データ発生装置１２のコードキャッシュデータの作成処理を説明する。

【００３１】乱数データ発生装置１２は、まずモデル情報ファイル４からコードキャッシュエントリ情報を読出し（ステップ３０１）、このエントリ情報に従ってハミングコード（ＥＣＣ）部分を作成し（ステップ３０２）、次に実際のコードデータを乱数発生させて乱数データを作成する（ステップ３０３）。

【００３２】さらに、コードキャッシュエントリのブロックの最後に割込み発生命令コードを埋め込み（ステップ３０４）、そして乱数データ設定装置１３を起動する（ステップ３０５）。

【００３３】次に、図４のフローによりハードウェア（Ｈ／Ｗ）レジスタのデータ作成／設定処理を説明する。

【００３４】もし、情報処理装置１の先行制御処理に待ちが生じた場合、つまりＬＤ（メモリからレジスタへのデータ転送）命令により確定するレジスタの内容を、次のＭＶＣ（メモリからメモリへのデータ転送）命令が、メモリアドレスとして使用している場合、このＬＤ命令の処理が終了するまでＭＶＣ命令のアドレス計算処理ができないために処理に待ちが生じてレジスタが一旦空き状態となる。

【００３５】このような状態が発生した場合、環境状態制御装置１１がたとえばＡステージレジスタでロックによりバブルが発生したことを感知し（ステップ４０１）、乱数データ発生装置１２に対してデータの生成指示を出す。

【００３６】さらに、乱数データ発生装置１２は、Ａステージレジスタのデータを作成し（ステップ４０２）、乱数データ設定装置１３に対してデータの設定指示を出す。

【００３７】そして、乱数データ設定装置１３は、Ａステージレジスタに生成された乱数値によるデータを埋め込む（ステップ４０３）。

【００３８】次に、図５により本実施例の論理検証環境制御装置を用いた論理検証の処理フローを説明する。

【００３９】まず、オペレータの指示により、環境状態制御装置１１が乱数データ発生装置１２を起動する（ステップ５０１）。そして、乱数データ発生装置１２は、モデル情報ファイル４から各ＲＡＭのエントリデータの生成情報を読み出し（ステップ５０２）、このモデル情報に従って各ＲＡＭのエントリデータを作成する（ステップ５０３）。

【００４０】さらに、乱数データ発生装置１２が乱数データ設定装置１３を起動し（ステップ５０４）、乱数データ設定装置１３がモデル情報ファイル４に登録されている各ＲＡＭのサイズに従って、作成されたエントリデータを各ＲＡＭに初期設定する（ステップ５０５）。

【００４１】そして、初期設定終了後、環境状態制御装置１１に制御を戻し（ステップ５０６）、この環境状態制御装置１１がオペレータの指示により該当する論理検証プログラムをメモリ１５にロードし（ステップ５０７）、この論理検証プログラムを起動する（ステップ５０８）。

【００４２】さらに、論理検証プログラムの実行が正常に終了すれば、環境状態制御装置１１がオペレータに対して次の処理要求をする処理を繰り返す。

【００４３】もし、論理検証プログラムが暴走した場合（ステップ５０９）、割込み発生命令によって制御が割込み制御装置１４に移り（ステップ５１０）、バックトレース可能なアドレスまで遡ってトレース情報を収集してトレース情報ファイル５に登録する（ステップ５１１）。

【００４４】そして、環境状態制御装置１１に制御が戻り（ステップ５１２）、環境状態制御装置１１がトレース情報を編集し、このトレース情報をシステムコンソール２に出力し（ステップ５１３）、これによって暴走原因の解析が行われる。

【００４５】従って、本実施例の論理検証環境制御装置によれば、環境状態制御装置１１、乱数データ発生装置１２、乱数データ設定装置１３および割込み制御装置１４が備えられ、エントリデータに乱数データを埋め込み、このエントリデータを情報処理装置１内の各ＲＡＭのメモリ１５、キャッシュ１６、レジスタ１７およびＴＬＢ１８などに初期設定することにより、論理検証の対象となる情報処理装置１内のデータ格納手段を、実際のユーザプログラムが実行されるのと等しい環境に設定することができ、これによって実使用環境下での論理検証が可能となる。

【００４６】また、論理検証プログラムが暴走した場合にも、割込み発生命令によって割込み発生アドレスからバックトレースが可能となるので、このトレース情報によって障害発生時の障害解析を容易に行うことができる。

【００４７】以上、本発明者によってなされた発明を実施例に基づき具体的に説明したが、本発明は前記実施例に限定されるものではなく、その要旨を逸脱しない範囲で種々変更可能であることはいうまでもない。

【００４８】たとえば、本実施例の論理検証環境制御装置については、論理検証の対象が情報処理装置１である場合について説明したが、本発明は前記実施例に限定されるものではなく、ゲートレベルの回路シミュレーションを実施する論理シミュレーション装置などについても広く適用可能である。

【００４９】また、乱数データ発生装置１２としては、たとえば特定の情報処理装置上で動作するような機械語命令コードおよびオペランドアドレスデータなどを発生するようにしてもよい。

【００５０】

【発明の効果】本願において開示される発明のうち、代表的なものによって得られる効果を簡単に説明すれば、下記のとおりである。

【００５１】(1).乱数データ発生装置、乱数データ設定装置、割込み制御装置および環境状態制御装置を備えることにより、エントリデータを論理フォーマットに整合するように作成し、このエントリデータに乱数データを埋め込んでデータ格納手段に設定することができるので、論理検証プログラムの実行以前に、論理シミュレーション装置または情報処理装置内資源を実使用環境と等価な環境に構築することができる。

【００５２】(2).乱数データ発生装置が特定の命令およびアドレスデータを発生することにより、特定の論理シミュレーション装置または情報処理装置上でのみ動作させることができる。

【００５３】(3).乱数データ発生装置が数十から数百バイトピッチに割込み発生命令オペランドコードを発生し、この割込み発生命令オペランドコードの割込み発生アドレスから割込み処理装置がバックトレースを実施してトレース情報をセーブすることができるので、障害が発生した場合の障害解析を容易に行うことができる。

【００５４】(4).前記(1)～(3)により、論理検証対象となる論理シミュレーション装置または情報処理装置に対して、実使用環境下での論理検証を可能とすることによって高精度な論理検証を実現すると共に、障害発生時の障害解析が容易に可能とされる論理検証環境制御装置を得ることができる。

【図面の簡単な説明】

【図１】本発明の一実施例である論理検証環境制御装置を示すブロック図である。

【図２】本実施例の論理検証環境制御装置における乱数データ発生装置および乱数データ設定装置の処理を示す説明図である。

【図３】本実施例において、乱数データ発生装置のキャッシュデータ作成処理を示すフローチャート図である。

【図４】本実施例において、乱数データ発生装置および乱数データ設定装置のハードウェアレジスタのデータ作成／設定処理を示すフローチャート図である。

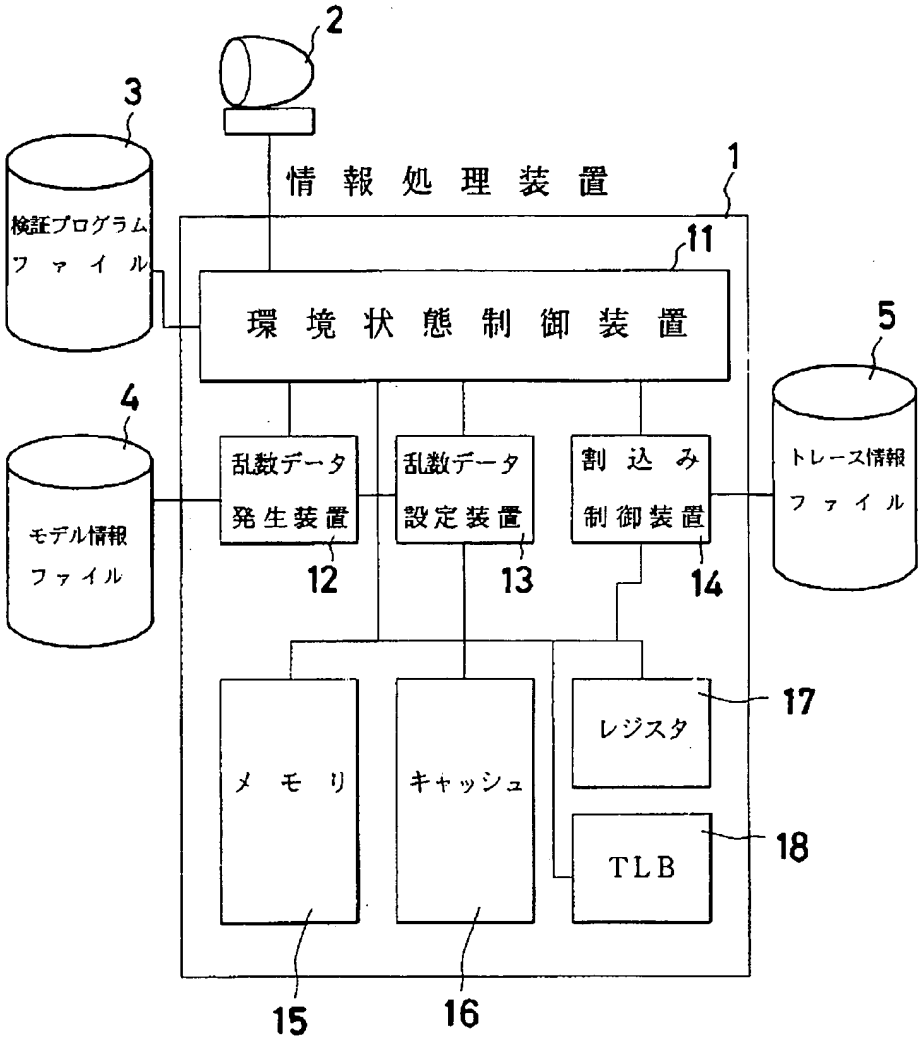
【図５】本実施例の論理検証環境制御装置を用いた論理検証処理を示すフローチャート図である。

【符号の説明】

１　情報処理装置
２　システムコンソール
３　検証プログラムファイル
４　モデル情報ファイル
５　トレース情報ファイル
１１　環境状態制御装置
１２　乱数データ発生装置
１３　乱数データ設定装置
１４　割込み制御装置
１５　メモリ
１６　キャッシュ
１７　レジスタ
１８　ＴＬＢ

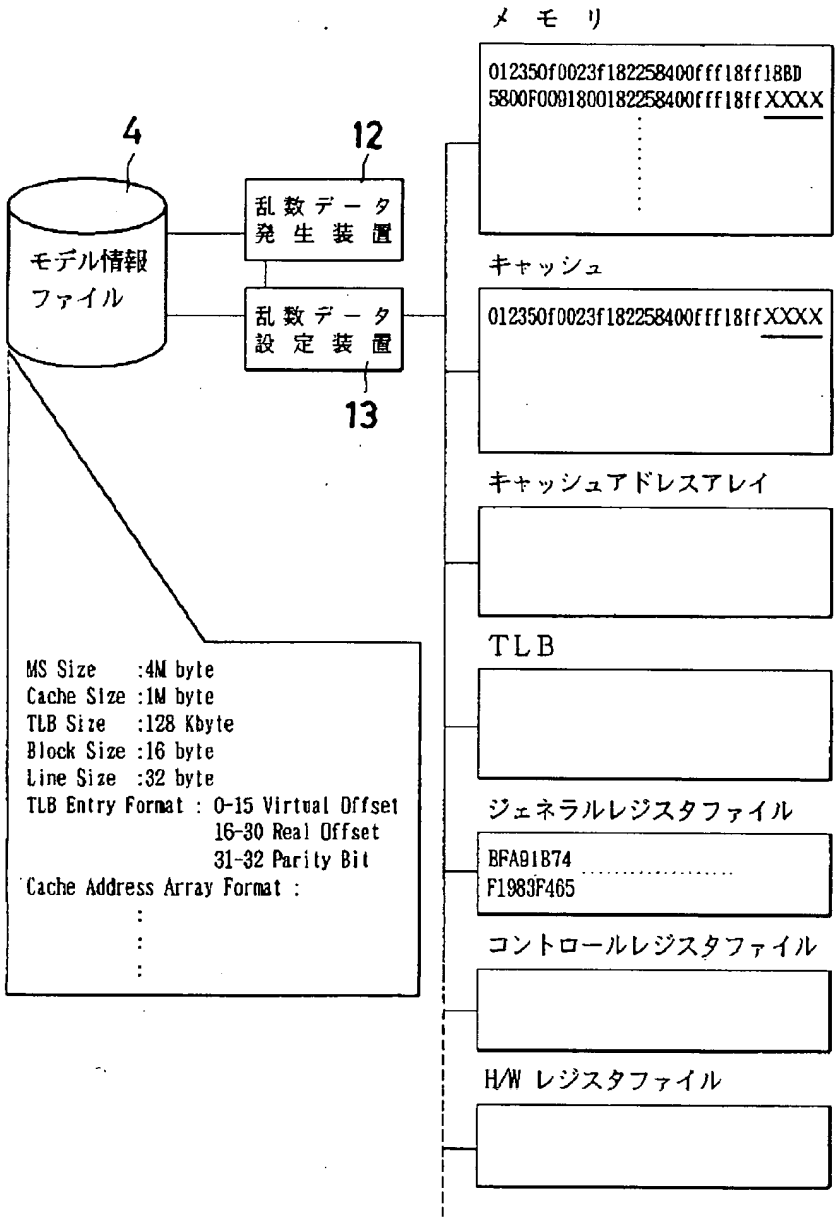
【図1】

図 1


2
3
検証プログラム
ファイル
情報処理装置
1
11
環境状態制御装置
5
4
モデル情報
ファイル
乱数データ
発生装置
乱数データ
設定装置
割込み
制御装置
トレース情報
ファイル
12
13
14
メモリ
キャッシュ
レジスタ
17
TLB
18
15
16

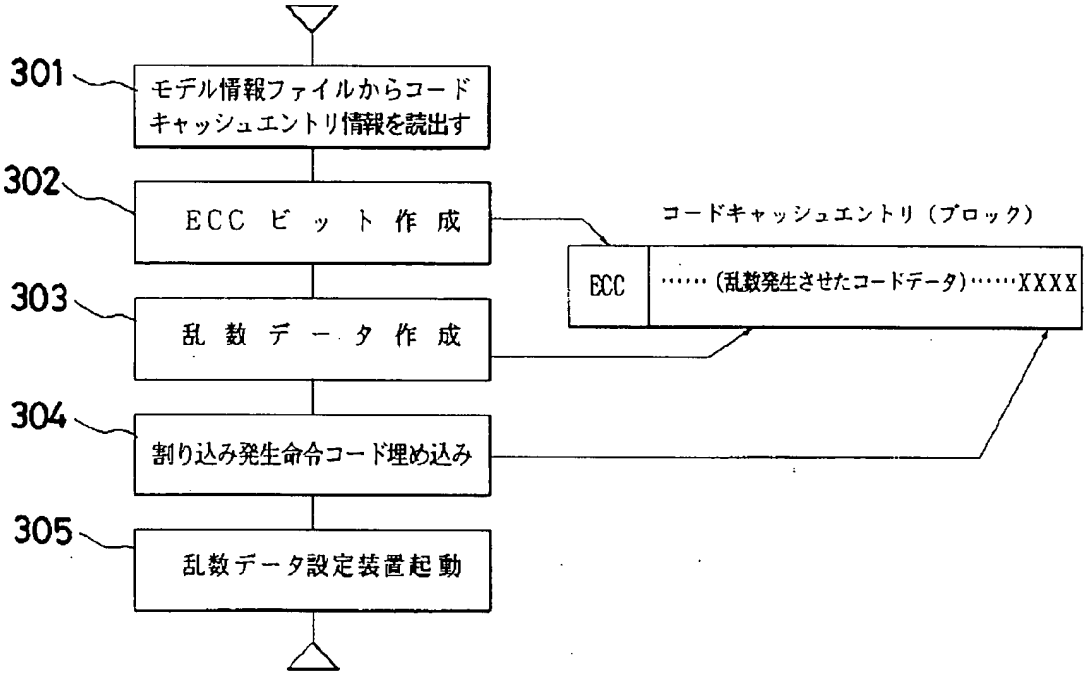
【図2】

図 2


メ モ リ
012350f0023f182258400fff18ff18BD
5800F00918001822584O0fff18ffXXXX
4
12
乱数データ発生装置
モデル情報ファイル
乱数データ設定装置
13
キャッシュ
012350f0023f182258400fff18ffXXXX
キャッシュアドレスアレイ
TLB
ジェネラルレジスタファイル
BFA91B74
F1983F465
コントロールレジスタファイル
H/W レジスタファイル
MS Size :4M byte
Cache Size :1M byte
TLB Size :128 Kbyte
Block Size :16 byte
Line Size :32 byte
TLB Entry Format : 0-15 Virtual Offset
16-30 Real Offset
31-32 Parity Bit
Cache Address Array Format :

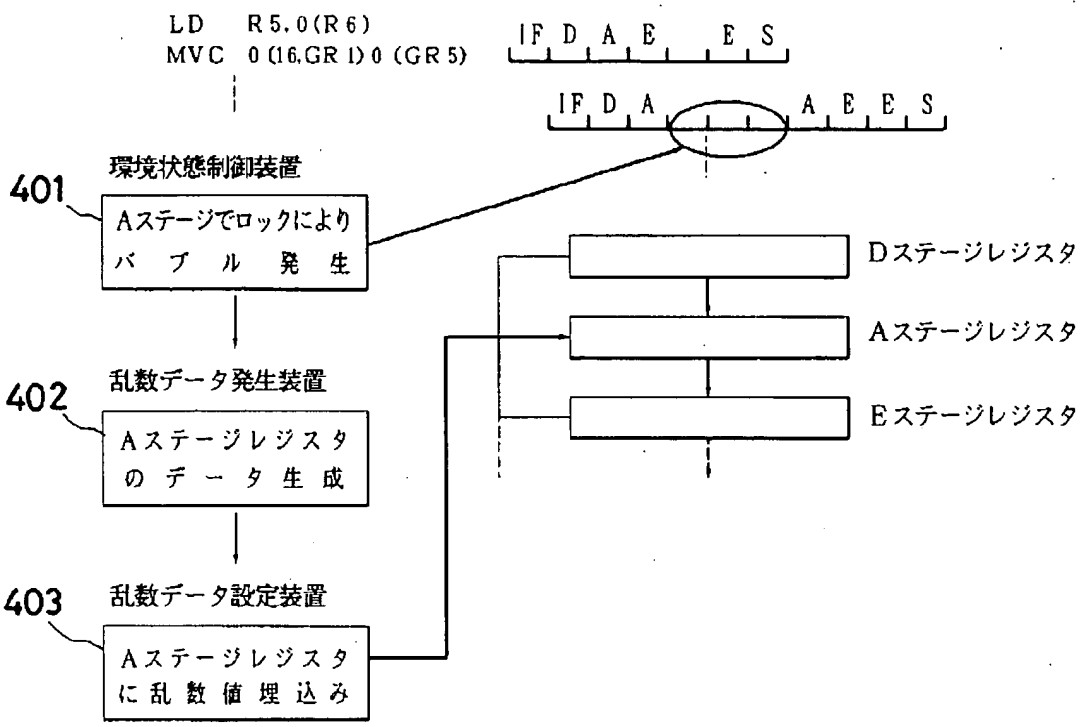
【図3】

図 3


301
モデル情報ファイルからコードキャッシュエントリ情報を読出す
302
ECC ビット作成
303
乱数データ作成
304
割り込み発生命令コード埋め込み
305
乱数データ設定装置起動
コードキャッシュエントリ（ブロック）
ECC
……（乱数発生させたコードデータ）……XXXX

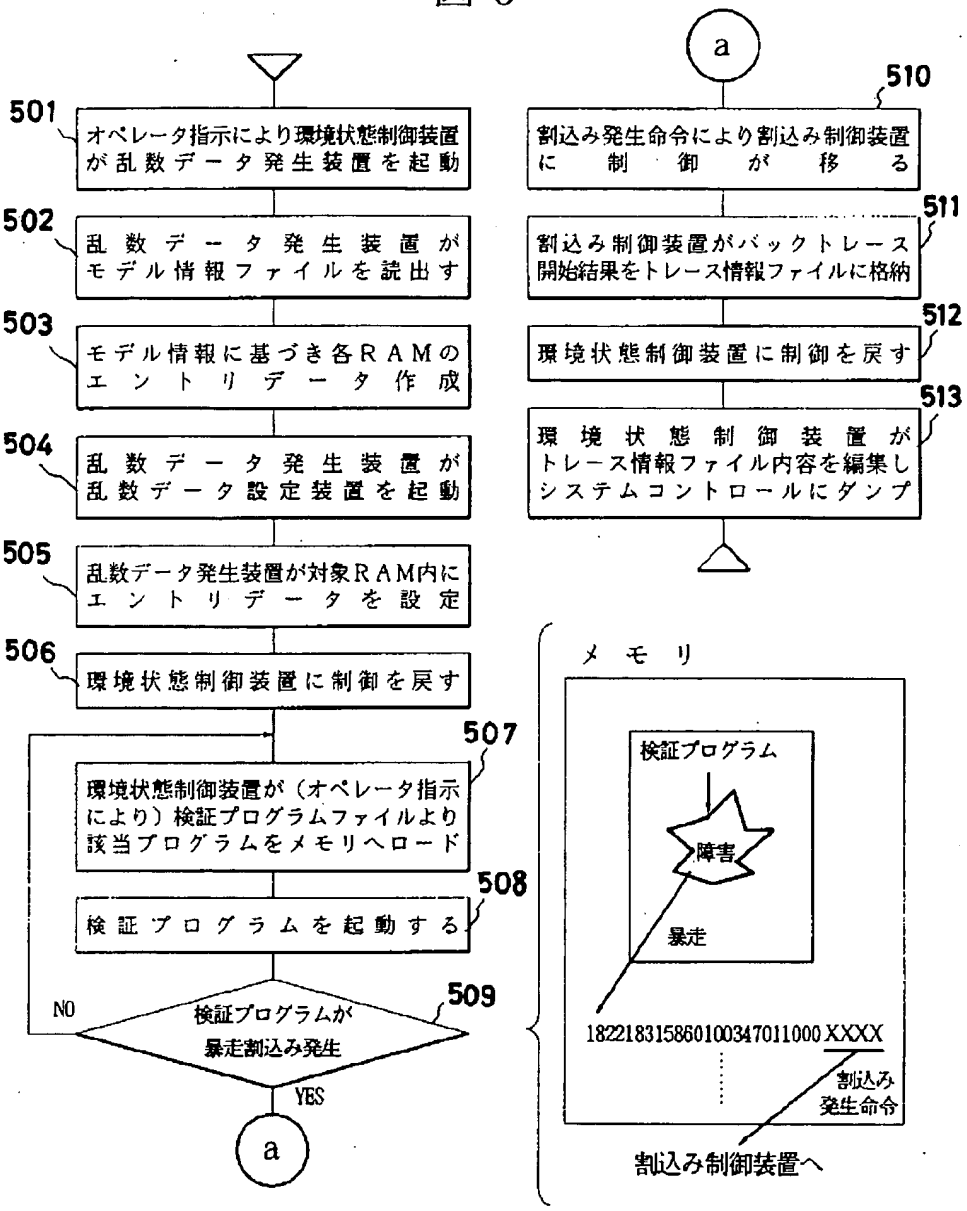
【図4】

図 4


LD　R5,0(R6)
MVC　0(16,GR1)0(GR5)
IF D A E E S
IF D A A E E S
環境状態制御装置
401
Aステージでロックにより
バ ブ ル 発 生
乱数データ発生装置
402
Aステージレジスタ
の デ ー タ 生 成
乱数データ設定装置
403
Aステージレジスタ
に乱数値埋込み
Dステージレジスタ
Aステージレジスタ
Eステージレジスタ

【図5】

図 5


501
オペレータ指示により環境状態制御装置が乱数データ発生装置を起動
502
乱数データ発生装置がモデル情報ファイルを読出す
503
モデル情報に基づき各RAMのエントリデータ作成
504
乱数データ発生装置が乱数データ設定装置を起動
505
乱数データ発生装置が対象RAM内にエントリデータを設定
506
環境状態制御装置に制御を戻す
507
環境状態制御装置が（オペレータ指示により）検証プログラムファイルより該当プログラムをメモリへロード
508
検証プログラムを起動する
509
検証プログラムが暴走割込み発生
NO
YES
a
a
510
割込み発生命令により割込み制御装置に制御が移る
511
割込み制御装置がバックトレース開始結果をトレース情報ファイルに格納
512
環境状態制御装置に制御を戻す
513
環境状態制御装置がトレース情報ファイル内容を編集しシステムコントロールにダンプ
メモリ
検証プログラム
障害
暴走
18221831586010034701100 0XXXX
割込み発生命令
割込み制御装置へ